

Program to read MI Extra-wide Aperture BPM test stand measurement data files, compute BPM transfer function in db difference formulation, and generate polynomial coefficients to compute position from traditional RF Module and Analog Box BPM System Outputs

Set working directory

```
CWD := "Z:\Instrumentation\MI_BPMs\ExtraWideApertureMIBPMs\EXWABPM_Raw_Data"
```

Import data:

Use data from Vertical measurements of all BPMs #1-8 and
Horizontal measurements of all BPMs #1-7

$$\begin{aligned} \text{hfilename} := & \begin{pmatrix} 12 \\ 22 \\ 32 \\ 42 \\ 5 \\ 6 \\ 7 \end{pmatrix} & \text{vfilename} := & \begin{pmatrix} 12 \\ 22 \\ 32 \\ 42 \\ 5 \\ 6 \\ 7 \\ 8 \end{pmatrix} \end{aligned}$$

Get horizontal and vertical scan data at orthogonally centered positions

```
hdatav0 := | a ← READPRN(concat("EXWA0", num2str(hfilename_0), "Horiz.txt"))
            mat ← submatrix(a, 169, 189, 0, 3)
            for i ∈ 1..length(hfilename) - 1
                | a ← READPRN(concat("EXWA0", num2str(hfilename_i), "Horiz.txt"))
                | mat ← stack(mat, submatrix(a, 169, 189, 0, 3))
            mat
```

$$\text{length}(\text{hdatav0}^{(0)}) = 147$$

```

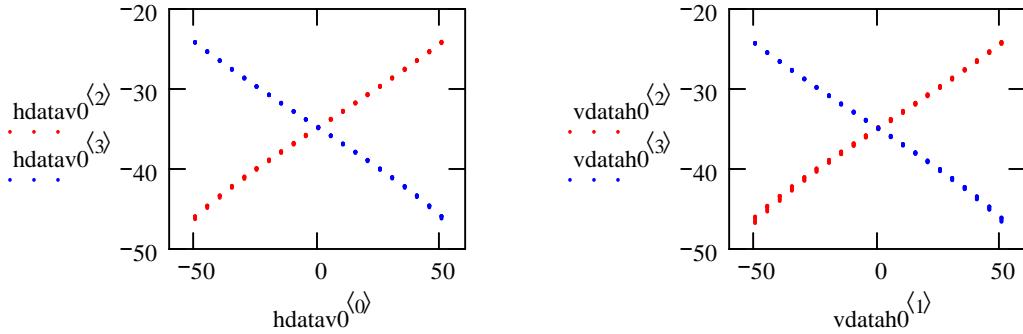
vdatah0 := | mrow ← 0
            | for i ∈ 0..length(vfilenum) - 1
            |   a ← READPRN(concat("EXWA0", num2str(vfilenumi), "Vert.txt"))
            |   for m ∈ 0..length(a⟨0⟩) - 1
            |     if |am,0| ≤ 0.5
            |       for c ∈ 0..3
            |         matmrow,c ← am,c
            |       mrow ← mrow + 1
            |
            | mat

```

$hdatav0bpm(k) := \text{submatrix}(hdatav0, \text{match}(k, hfilenum)_0 \cdot 21, \text{match}(k, hfilenum)_0 \cdot 21 + 20, 0, 3)$
 $hbpm(k) := hdatav0bpm(k)$

$vdatah0bpm(k) := \text{submatrix}(vdatah0, \text{match}(k, vfilenum)_0 \cdot 21, \text{match}(k, vfilenum)_0 \cdot 21 + 20, 0, 3)$
 $vbpm(k) := vdatah0bpm(k)$

$$\text{length}(vdatah0^{⟨0⟩}) = 168$$

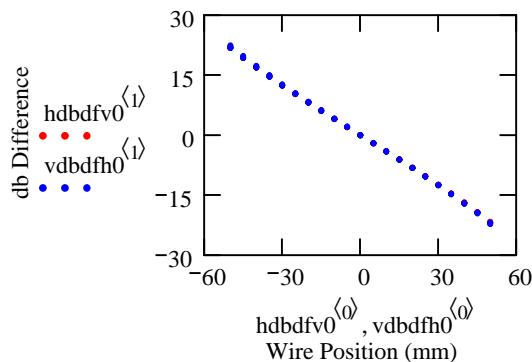


Plots of electrode signal amplitudes in dB vs. wire position in millimeters.
Horizontal BPM scans in left plot and vertical scans in right.

Now create two-column arrays for Horizontal and Vertical, first column 'in-plane' position and second column db difference

```
hdbdfv0 := | result $\langle 0 \rangle$  ← hdatav0 $\langle 0 \rangle$ 
             | for m ∈ 0..rows(hdatav0) - 1
             |   resultm, 1 ← hdatav0m, 3 - hdatav0m, 2
             |
             | result
```

```
vdbdfh0 := | result $\langle 0 \rangle$  ← vdatah0 $\langle 1 \rangle$ 
             | for m ∈ 0..rows(vdatah0) - 1
             |   resultm, 1 ← vdatah0m, 3 - vdatah0m, 2
             |
             | result
```



Now generate 5th order polynomial fits to this data

```
oncenterfith := regress(hdbdfv0 $\langle 0 \rangle$ , hdbdfv0 $\langle 1 \rangle$ , 5)
oncenterfitv := regress(vdbdfh0 $\langle 0 \rangle$ , vdbdfh0 $\langle 1 \rangle$ , 5)
inversefith := regress(hdbdfv0 $\langle 1 \rangle$ , hdbdfv0 $\langle 0 \rangle$ , 5)
inversefitv := regress(vdbdfh0 $\langle 1 \rangle$ , vdbdfh0 $\langle 0 \rangle$ , 5)
```

These are coefficients for db = poly(mm)
Ignore first 3 values in column!

$$\text{oncenterfith} = \begin{pmatrix} 3 \\ 3 \\ 5 \\ -0.027 \\ -0.406 \\ 1.263 \times 10^{-5} \\ -8.998 \times 10^{-6} \\ 5.387 \times 10^{-9} \\ -1.354 \times 10^{-9} \end{pmatrix} \quad \text{oncenterfitv} = \begin{pmatrix} 3 \\ 3 \\ 5 \\ -0.037 \\ -0.407 \\ 1.809 \times 10^{-5} \\ -9.696 \times 10^{-6} \\ -6.801 \times 10^{-9} \\ -1.483 \times 10^{-9} \end{pmatrix}$$

These are coefficients for mm = poly(db)

$$\text{inversefith} = \begin{pmatrix} 3 \\ 3 \\ 5 \\ -0.068 \\ -2.467 \\ 2.53 \times 10^{-4} \\ 3.572 \times 10^{-4} \\ 1.037 \times 10^{-7} \\ 2.737 \times 10^{-8} \end{pmatrix} \quad \text{inversefitv} = \begin{pmatrix} 3 \\ 3 \\ 5 \\ -0.087 \\ -2.458 \\ 1.335 \times 10^{-4} \\ 3.612 \times 10^{-4} \\ -1.05 \times 10^{-7} \\ 5.821 \times 10^{-8} \end{pmatrix}$$

For further transfer function computations set offset term to zero and use:

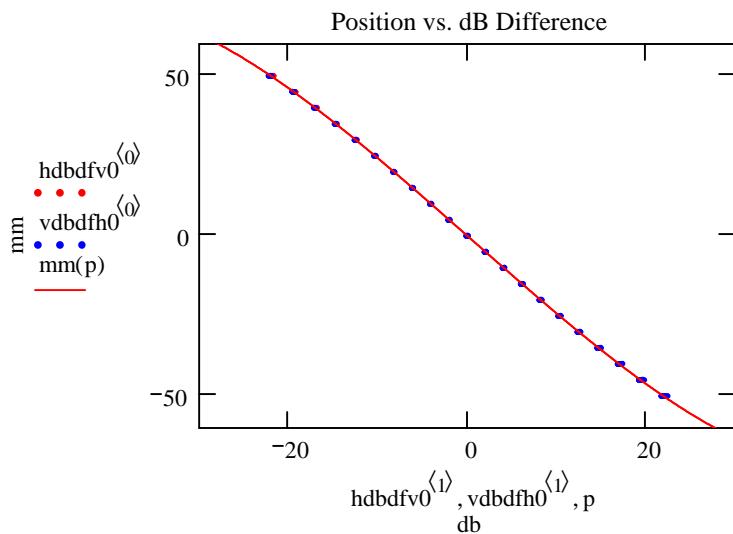
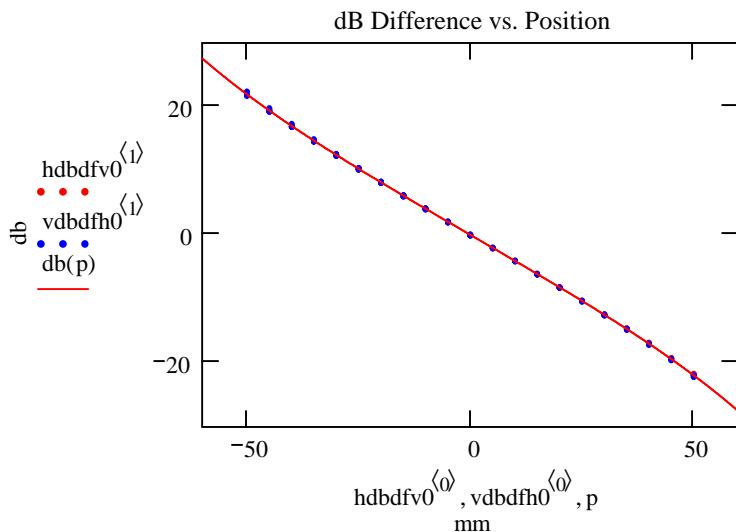
$$\text{mm2db} := \begin{pmatrix} 0 \\ -0.4065 \\ 0 \\ -9.3 \times 10^{-6} \\ 0 \\ -1.483 \times 10^{-9} \end{pmatrix} \quad \text{db(mm)} := \sum_{i=0}^5 [(\text{mm2db}_i) \cdot \text{mm}^i]$$

And

$$db2mm := \begin{pmatrix} 0 \\ -2.46 \\ -1.9 \times 10^{-4} \\ 3.6 \times 10^{-4} \\ 0 \\ 4 \times 10^{-8} \end{pmatrix}$$

$$mm(db) := \sum_{i=0}^5 \left[(db2mm_i) \cdot db^i \right]$$

And check polynomial function against data



So now have polynomial BPM transfer function between dB and millimeters

Need path to compute millimeters from output of RF Module and Analog Box digitizer system

Work backwards from Analog Box digitizer output value N and use system equations from Jim Crisp's note " Main Injector bpm scale factors" in Beams Document #1344-v1

First compute table of Vrfmod vs. N for Analog Box digitizers

$$n := 0..255 \quad N_n := n$$

$$Vrfmod_n := \frac{(128 - N_n)}{57.41}$$

$$\text{Table} := \text{augment}(N, Vrfmod)$$

Now, using these discrete values of Vrfmod, compute corresponding RF Module input difference in dB and then use the BPM transfer function to find corresponding position in millimeters

$$F := 1.14 \quad C1 := 0.2974$$

$$dbin(vrfmodout) := \frac{-20}{F \cdot \ln(10)} \cdot \ln\left(\tan\left(F \cdot C1 \cdot vrfmodout + \frac{\pi}{4}\right)\right)$$

$$dbinput_n := dbin(Vrfmod_n)$$

$$mmin_n := mm(dbinput_n)$$

Add dbinput and position columns to "Table" array so that it now has columns N, Vrfmod, dbdiffin, mmEWABPM

$$\text{Table} := \text{augment}(\text{Table}, dbinput, mmin)$$

Create BPM_DaughterCard_Table having only columns N and mmEWABPM

$$\text{BPM_DaughterCard_Table}^{(0)} := \text{Table}^{(0)}$$

$$\text{BPM_DaughterCard_Table}^{(1)} := \text{Table}^{(3)}$$

Write BPM_DaughterCard_Table to File. This is the file to be used to "look-up" beam position given a BPM daughter card digitizer output value

```
CWD := "Z:\Instrumentation\MI_BPMs\ExtraWideApertureMIBPMs"
```

```
WRITEPRN("EWA_BPM_DaughterCard_Table.txt") := BPM_DaughterCard_Table
```

Now using Vrfmod and mmEWABPM values from "Table" array find fit between Vrfmod and position

```
vrfmo := regress(Table<1>,Table<3>,5)
```

$$vrfmo = \begin{pmatrix} 3 \\ 3 \\ 5 \\ 0.019 \\ 13.832 \\ -0.057 \\ -0.721 \\ 0.014 \\ 0.592 \end{pmatrix}$$

Now define K vector as polynomial coefficients to compute position from Vrfmod with offset term set to zero

```
K := submatrix(vrfmo,3,8,0,0)
K_0 := 0
```

$$K = \begin{pmatrix} 0 \\ 13.832 \\ -0.057 \\ -0.721 \\ 0.014 \\ 0.592 \end{pmatrix}$$

$$\text{posmm}(vrfm) := \sum_{i=0}^5 \left(K_i \cdot vrfm^i \right)$$

Finally get polynomial function for position channel MADC database entries. MADC input voltage is "vdac" which is 2.251 times the RF module output voltage.

$$\text{posdacmm}(vdac) := \left[\sum_{i=0}^5 \left[K_i \left(\frac{vdac}{2.251} \right)^i \right] \right]$$

Define Kmadc as polynomial coefficients for computing position from MADC input.

$$\text{m} := 0..5 \quad \text{Kmadc}_m := \frac{K_m}{2.251^m}$$

$$\text{Kmadc} = \begin{pmatrix} 0 \\ 6.145 \\ -0.011 \\ -0.063 \\ 5.386 \times 10^{-4} \\ 0.01 \end{pmatrix} \quad \text{posmadcmm(vdac)} := \left[\sum_{i=0}^5 \left(\text{Kmadc}_i \cdot \text{vdac}^i \right) \right]$$

$$rfv := 2.3$$

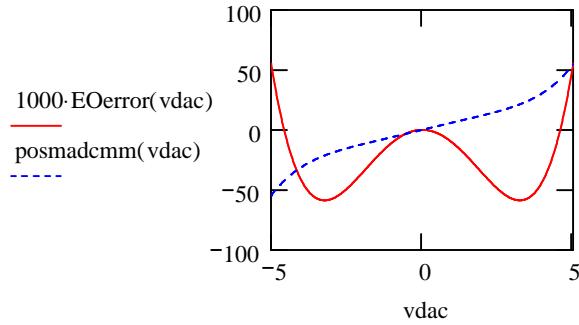
$$\text{madcvolts} := rfv \cdot 2.251$$

$$\text{madcvolts} = 5.177$$

$$\text{posmm}(rfv) = 61.201 \quad \text{posmadcmm}(\text{madcvolts}) = 61.201$$

Check significance of even terms

$$\text{EOerror}(vdac) := \text{posmadcmm}(vdac) - \sum_{i=0}^2 \left(\text{Kmadc}_{2 \cdot i + 1} \cdot \text{vdac}^{2 \cdot i + 1} \right)$$



See only 50 um errors out to 50 mm radius if neglect even order terms so neglect them.

Summary and Conclusion

A lookup table to find beam position in millimeters given BPM daughter card digitizer value for the MI ExtraWide Aperture BPMs has been created and written as file "EWA_BPM_DaughterCard_Table.txt".

A sufficiently accurate polynomial for computing position in mm from MADC volts for the EWA BPMs is:

$$mm = 6.145 * V_{madc} - 0.063 * V_{madc}^{**3} + 0.01 * V_{madc}^{**5}$$